

# Bearing Stabilization and Tracking for an AUV with an Acoustic Line Array

Andrew J. Poulsen and Donald P. Eickstedt  
Massachusetts Institute of Technology  
Cambridge, MA 02139  
Email: poulsen@mit.edu; eicksted@mit.edu

John P. Ianniello  
SAIC  
Mystic, CT 06355  
Email: john.p.ianniello@saic.com

**Abstract**—Passive underwater detection and tracking sonar systems using autonomous underwater vehicles (AUVs) have many important applications. Because of imperfections in vehicle control, it is common for an AUV to undergo significant yaw and pitch oscillations. As a result, it is important to compensate for the vehicle motion when generating true bearing estimates while using a rigidly attached acoustic line array. This paper describes full beam interpolation tracker and bearing stabilization algorithms that were implemented to address these issues on an intelligent AUV sonar sensor and tested during a subsequent sea trial with the goal of providing target bearing estimates to a target track estimation algorithm. These beam tracking and bearing stabilization algorithms can also be applied to the case of a flexible towed array with some additional modifications. Initial results indicate that this is an effective method of measuring stabilized true target bearings.

## I. INTRODUCTION

Underwater detection and tracking systems have many important applications. Targets of interest may consist of either man-made or natural sources. An investigation is being undertaken at MIT to use a network of autonomous underwater vehicles (AUVs) with acoustic line arrays to adaptively and cooperatively track moving underwater targets. Each AUV is equipped with an intelligent sonar sensor which passively acquires acoustic data, detects acoustically active targets, and provides target track information to a behavior-based vehicle control system which attempts to maneuver the AUV to improve the target track estimation. Due to imperfections in vehicle control, however, it is common for an AUV to undergo significant yaw and pitch oscillations. In this scenario, it is necessary to compensate for the vehicle motion in order to generate accurate true bearing estimates.

The solution we propose is based on a modified version of a full beam tracking algorithm generally available in the sonar community [1]. It is particularly relevant to a rigidly attached linear acoustic pressure sensor array mounted on a sensing platform. The acoustic sensor in use at MIT consists of a "swordfish" linear array mounted to the nose cone of an AUV (see Fig. 1). Such an array moves directly with the vehicle pitch and yaw, making it necessary to compensate for this motion. For a flexible towed acoustic array, on the other hand, the array orientation is not as susceptible to vehicle pitch and yaw oscillations, and requires an entirely different type of array orientation model.

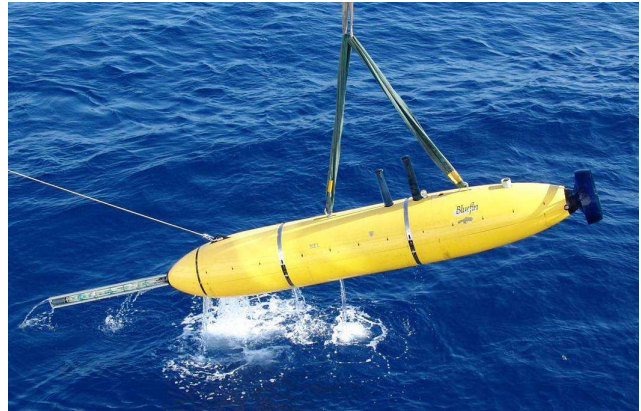


Fig. 1. AUV with mounted "swordfish" linear hydrophone array used during at sea trials to obtain the results presented in Section IV.

Because of limited computational assets, we chose to implement a conventional (rather than adaptive) beamformer. In addition to their computational advantages, conventional beamformers provide the benefit of a predictable main lobe shape, which is constant with steering angle in  $\cos \theta$  space. This is important for reliable beam interpolation. Here,  $\theta$  refers to the conical scanning angle of a uniformly spaced linear array.

A simple energy detector was implemented, which is suitable for high signal to noise ratio environments. The detector initially samples and estimates the ambient noise mean and standard deviation. The detector then uses a threshold above the estimated noise floor to declare a detection. This threshold is a function of both the estimated mean and standard deviation.

With a linear pressure sensor array, only conical angles can be directly measured. After each update cycle, the conical angle is measured by selecting the beam with the highest measured power out of a fixed number of beams (spaced uniformly in  $\cos \theta$  space), interpolating using the power levels of the two adjacent beams, and applying a bias correction. This provides bearing resolution while only requiring computation of a relatively low number of beams. An error signal, defined as the difference between the latest measurement and the last smoothed conical angle estimate, is used to update both the

smoothed conical angle and angle rate estimates via an alpha-beta filter. Subsequently, using the geometrical relationship between conical angle, relative bearing, pitch, and vertical arrival angle (which must be either estimated separately or assumed a priori), the stabilized relative bearing is computed. The stabilized true bearing is then computed by adding the current AUV heading to the relative bearing. Additional details are provided in Sec. III.

The beamforming and bearing stabilization algorithms were integrated into a logical target tracking sensor developed for integration with a behavior-based AUV control system, the goal of which is to provide adaptive, autonomous tracking of moving underwater targets. Several levels of cooperation between the logical tracking sensor and the vehicle control system were developed. First, tracking only begins after detection is declared. Upon detection, it is necessary to resolve any right/left ambiguity by deciding which side of the array the contact is on. This is accomplished by changing the AUV's course and noting whether the relative bearing moves forward or aft. After this is completed, stabilized bearing tracking begins. An additional level of autonomy is that the bearing tracker must determine if the contact has changed sides when it moves into and out of an endfire beam.

In the sections that follow, we describe the technical approach used to implement the bearing tracker on the intelligent sensor platform (Sec. II), details about the beamformer and stabilized bearing tracking algorithm (Sec. III), followed by results from a sea trial (Sec. IV).

## II. TECHNICAL APPROACH

In this section, we present our general autonomy architecture and how the particular components that reflect the contribution of this work fit into that architecture. The outline for experimental validation is also discussed.

### A. The MOOS-IvP Autonomy Architecture

The AUVs described in this work use the MOOS-IvP architecture for autonomous marine vehicle control. MOOS-IvP is composed of the Mission Oriented Operating Suite (MOOS), an open source software project for coordinating software processes running on an autonomous platform, typically under GNU/Linux. MOOS-IvP also contains the IvP Helm, a behavior-based helm that runs as a single MOOS process and uses multi-objective optimization with the Interval Programming (IvP) model for behavior coordination [2], [3]. See [4] and [5] for other examples of MOOS-IvP on autonomous marine vehicles.

A MOOS community contains processes that communicate through a database process called the MOOSDB, as shown in Fig. 2(a). MOOS ensures a process executes its "Iterate" method at a specified frequency and handles new mail on each iteration in a publish and subscribe manner. The IvP Helm runs as the MOOS process pHelmIvP (Fig. 2(b)). Each iteration of the helm contains the following steps: (1) mail is read from the MOOSDB, (2) information is updated for consumption by behaviors, (3) behaviors produce an objective function if

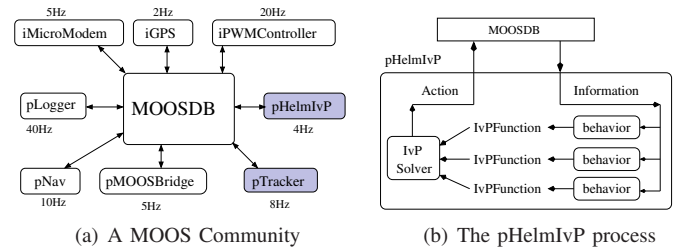


Fig. 2. The IvP Helm runs as a process called pHelmIvP in a MOOS community. MOOS may be composed of processes for data logging (pLogger), data fusion (pNav), actuation (iPWMController), sensing (iGPS), communication (pMOOSBridge, iMicroModem), and much more. They can all be run at different frequencies as shown.

applicable, (4) the objective functions are resolved to produce a single action, and (5) the action is posted to the MOOSDB for consumption by low-level control MOOS processes.

### B. The Logical Sonar Sensor

The logical sonar sensor (see Fig. 3) consists of the physical acoustic sampling hardware as well as algorithms that abstract the real-time data into higher forms of information suitable for a behavior-based control system. Because of the distributed MOOS architecture, the actual sensor and processing algorithms (MOOS processes) may well reside in a separate vehicle payload from the main vehicle control computer [6]. The tracking vehicles in this work use a set of target tracking algorithms that run in a single MOOS process called pTracker (see Fig. 2(a)). This process subscribes to target bearing data from the MOOS database. The bearing data is either produced by another MOOS process interfaced with a physical bearings-only sensor, or the bearing data is produced by an alternative MOOS process that simulates bearings-only sensor data. In this work, the beam tracking and stabilization algorithms described in detail in Sec. III are part of a MOOS process called pBearings which uses raw element-level hydrophone data to produce the stabilized bearings which are then placed in the MOOS database for consumption by pTracker. The pTracker process then produces and posts track solution information to the MOOSDB to be consumed by any other MOOS process including inter-vehicle communications processes like pMOOSBridge or iAcousticModem or the behaviors in the vehicle control system. Feedback from the platform behaviors is available for dynamically changing the sensor parameters in response to the platform state. More information on the target tracking algorithms and vehicle behaviors for the adaptive tracking scenario can be found in [7].

### C. Validation with Experimental Data

Experimental validation of this work is presented using an Odyssey-III AUV equipped with a passive acoustic payload and nose-mounted linear hydrophone array. The beamforming and stabilized bearing tracking algorithms were coded and debugged in MATLAB, translated into C++, implemented on the intelligent sonar sensor, and tested during a sea trial in July, 2005 off the coast of Pianosa, Italy.

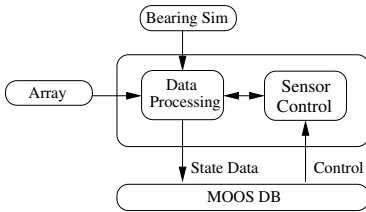


Fig. 3. The logical sonar sensor. Rather than passing raw acoustic data directly to the platform control system, the sensor processes the acoustic data into a higher level of abstraction suitable for a behavior-based control system. Feedback from the platform behaviors is available for dynamically changing the sensor parameters in response to the platform state.

### III. DESCRIPTION OF BEAMFORMER AND STABILIZED TRUE BEARING TRACKING ALGORITHM

In this section, we describe in detail the implementation of the beamformer and stabilized true bearing algorithm. The raw hydrophone array data is processed by a conventional beamformer which generates a spatial power spectrum at a discrete number of beams or look directions. The bearing stabilization and tracking algorithm uses this information to generate the desired output: the true azimuth bearing estimate of the contact.

#### A. Beamformer

Due to limitations of available on-board, real-time computational assets, we chose to implement a broadband conventional (rather than adaptive) beamformer with several tunable parameters. These include processing bandwidth, temporal window overlap percentage, windowing functions (both in time and across frequencies), discrete Fourier transform (DFT) size, number of beams  $N$ , among others. In addition to their computational advantages, conventional beamformers provide the benefit of a predictable main lobe shape, which is constant with steering angle in  $\cos\theta$  space for a uniformly spaced linear array. This is important for reliable beam interpolation. A cosine spaced beamformer is used to ensure that the interpolation process is independent of the actual conical angle of the contact.

The beamformer generates the power spectrum vector  $\mathbf{P}_n$  which is the spatial power spectrum sampled at  $N$  discrete points (beams),

$$\mathbf{P}_n = \begin{bmatrix} p_n(u_1) \\ \vdots \\ p_n(u_k) \\ \vdots \\ p_n(u_N) \end{bmatrix} \quad (1)$$

where the index  $k = 1, \dots, N$  specifies the beam number. Note that  $n$  is a discrete time index,  $u_k = \cos\theta_k$  is the  $k^{\text{th}}$  beam angle in  $\cos\theta$  space and  $\theta_k$  is the conical angle of the  $k^{\text{th}}$  beam. The elements of the spatial power spectrum  $\mathbf{P}_n$  are given by

$$p_n(u_k) = \sum_l |\mathbf{x}_n(\omega_l)^H \mathbf{v}(\theta_k, \omega_l)|^2 \quad (2)$$

where  $(\cdot)^H$  is the conjugate transpose operator and  $\mathbf{x}_n(\omega_l)$  is the windowed array data snapshot vector corresponding to the  $l^{\text{th}}$  frequency DFT bin (represented by  $\omega_l$ ) at time index  $n$ . The array manifold vector  $\mathbf{v}(u_k, \omega_l)$  corresponds to the  $l^{\text{th}}$  frequency bin and  $k^{\text{th}}$  conical beam. The summation over  $l$  in (2) is over the DFT frequency bins of interest corresponding to the acoustic energy radiated by the target.

#### B. Beam Interpolation, Bearing Stabilization and Tracking

As mentioned previously, a full beam interpolation tracker was implemented. The algorithm used is based on a modified version of a full beam tracker generally available in the sonar community [1]. The tracker receives the beam power spatial spectrum  $\mathbf{P}_n$  at each beamformer time update, then selects the beam with the largest power. Denote the beam number of this beam as

$$k_{\max,n} = \arg \max_k p_n(u_k). \quad (3)$$

The maximum beam power is therefore represented by

$$p_{\max,n} = p_n(u_{k_{\max,n}}) \quad (4)$$

where  $u_{k_{\max,n}}$  is the cosine of the conical angle of the beam corresponding to  $k_{\max,n}$ . Note that in this section, many of the expressions contain a time index suffix (such as  $n$ ). This is either to emphasize that an expression must be updated each time through the cycle and/or to distinguish between separate instances of the same expression with different time indices.

The algorithm first verifies that the beam with maximum power is not one of the two endfire beams. If not, then using the power of the beams adjacent to beam  $k_{\max,n}$ , it computes a crude estimate of a fractional change in the beam number, represented by  $\gamma_{1,n}$ , using the formula

$$\gamma_{1,n} = -\frac{p_{\text{right},n} - p_{\text{left},n}}{\max\{p_{\max,n} - p_{\text{left},n}, p_{\max,n} - p_{\text{right},n}\}}. \quad (5)$$

Here, the adjacent right and left beam powers are respectively represented by  $p_{\text{right},n}$  and  $p_{\text{left},n}$ , where

$$p_{\text{right},n} = p_n(u_{(k_{\max,n}+1)}) \quad (6)$$

and

$$p_{\text{left},n} = p_n(u_{(k_{\max,n}-1)}). \quad (7)$$

Note that  $\gamma_{1,n} \in [-1, 1]$  and that  $u_{k_{\max,n}} + \gamma_{1,n}\Delta u/2$  is a crude estimate of the cosine of the conical angle of the contact, where  $\Delta u$  is the spacing between the beams in conical angle, such that

$$\Delta u = \frac{2}{N-1}. \quad (8)$$

The form in (5) is used instead of a standard parabolic interpolation form since it has been observed to have lower variance [1]. Note that, as is also the case with parabolic interpolation, the estimate is biased unless  $p_{\text{right},n} = p_{\text{left},n}$  (the true bearing estimate is coincident with one of the beams), or unless  $p_{\max,n} = p_{\text{left},n}$  or similarly  $p_{\max,n} = p_{\text{right},n}$  (the true bearing estimate is halfway between two beams). To decrease the estimate bias, the estimate  $\gamma_{1,n}$  from (5) is improved using an iterative procedure. This bias corrected estimate, denoted

by  $\gamma_{2,n}$ , is then incorporated with  $u_{k_{\max},n}$  to yield the current measurement of the cosine of the conical angle of the contact for time increment  $n$ ,

$$\hat{u}_n = u_{k_{\max},n} + \frac{\gamma_{2,n}\Delta u}{2}. \quad (9)$$

Next, a preliminary estimate of the stabilized relative horizontal azimuth bearing to the contact ( $\tilde{\theta}_{\text{rel},n}$ ) is computed by taking the difference of the last estimated true bearing  $\theta_{\text{true},n-1}$  and the current heading of the array  $\theta_{\text{head},n}$ , i.e.

$$\tilde{\theta}_{\text{rel},n} = \theta_{\text{true},n-1} - \theta_{\text{head},n}. \quad (10)$$

This is then used to make a preliminary estimate of the stabilized cosine of the conical angle, computed using the geometrical relation

$$\tilde{u}_n = \cos(\tilde{\theta}_{\text{rel},n}) \cos(\psi_n) \cos(\phi) + \sin(\psi_n) \sin(\phi) \quad (11)$$

where  $\psi_n$  is the current measured pitch of the array and  $\phi$  is the vertical angle of arrival of the contact signal (generally estimated separately or assumed a priori). Note that the estimates of  $\tilde{\theta}_{\text{rel},n}$  and  $\tilde{u}_n$  do not take into account the beamforming information from the current time index  $n$ . They do, however, carry the suffix  $n$  since they incorporate the current array heading measurement at time index  $n$ .  $\tilde{\theta}_{\text{rel},n}$  and  $\tilde{u}_n$  are distinguished from other estimates computed in (9), (14) and (17) by the  $\sim$  symbol.

An error signal is formed by taking the difference of (9) and (11),

$$e_n = \hat{u}_n - \tilde{u}_n \quad (12)$$

and  $\tilde{u}_n$  is filtered by an alpha-beta filter to produce an improved estimate of the cosine of the conical angle of the contact for time increment  $n$ ,  $u_{\text{filt},n}$ . This alpha-beta filter [8] is given by

$$\dot{u}_n = \dot{u}_{n-1} + \frac{\beta e_n}{\Delta t} \quad (13)$$

$$u_{\text{filt},n} = \tilde{u}_n + \dot{u}_n \Delta t + \alpha e_n. \quad (14)$$

Here,  $\Delta t$  is the update time increment;  $\dot{u}_{n-1}$  and  $\dot{u}_n$  are the previous and updated angle rate estimates, respectively; and  $\alpha$  and  $\beta$  are the alpha-beta filter coefficients.  $\beta$  is computed from  $\alpha$  via

$$\beta = \frac{\alpha^2}{2 - \alpha} \quad (15)$$

which gives the ‘‘optimal’’  $\beta$  [8]. In addition,  $\alpha$  and  $\beta$  are chosen to yield the desired effective averaging time (in update cycles) given by

$$T_{\text{eff}} = \frac{\alpha(4 - \alpha - 2\beta)}{2\alpha^2 + \beta(2 - 3\alpha)}. \quad (16)$$

The current smoothed relative bearing is then computed from

$$\theta_{\text{rel},n} = \text{atan2}(Y, X) \quad (17)$$

where  $\text{atan2}(\cdot, \cdot)$  is the four quadrant inverse tangent, with

$$X = \frac{u_{\text{filt},n} - \sin(\phi) \sin(\psi_n)}{\cos(\phi) \cos(\psi_n)} \quad (18)$$

and

$$Y = \eta_{\text{PS}} \sqrt{1 - X^2}. \quad (19)$$

The flag  $\eta_{\text{PS}}$  indicates whether the target is on the port or starboard. The true bearing is then updated via

$$\theta_{\text{true},n} = \theta_{\text{rel},n} + \theta_{\text{head},n}. \quad (20)$$

This completes the basic cycle as long as the contact does not enter an endfire beam.

If the contact does enter an endfire beam, an acceptable updated estimate of the conical angle cannot be made so the true bearing is held at the last estimated value

$$\theta_{\text{true},n} = \theta_{\text{true},n-1} \quad (21)$$

which tacitly assumes that the true bearing is not changing rapidly. Note that  $\tilde{\theta}_{\text{rel},n}$  is updated as in (10) in this scenario.

Another check is made during each iteration to determine if the array is turning and if  $\tilde{\theta}_{\text{rel},n}$  is closer than a selected value,  $\theta_{\text{min}}$ , of endfire. If so, this indicates that the contact is passing into endfire due to the array turning. The value of true bearing is again held at the last good value as in (21). While the contact is in endfire,  $\eta_{\text{PS}}$  is re-computed during each time cycle as

$$\eta_{\text{PS}} = \text{sign} \left\{ \sin(\tilde{\theta}_{\text{rel},n}) \right\}. \quad (22)$$

When the contact exits endfire, the bearing tracker returns to ‘‘normal’’ operation.

To determine if the array is turning, it is necessary to filter the headings in another alpha-beta filter and determine if the turn rate exceeds a selected threshold. The algorithm as it stands does not correctly address the case where the contact enters and exits an endfire beam without the array turning. This could be accounted for, however, with more complicated logic.

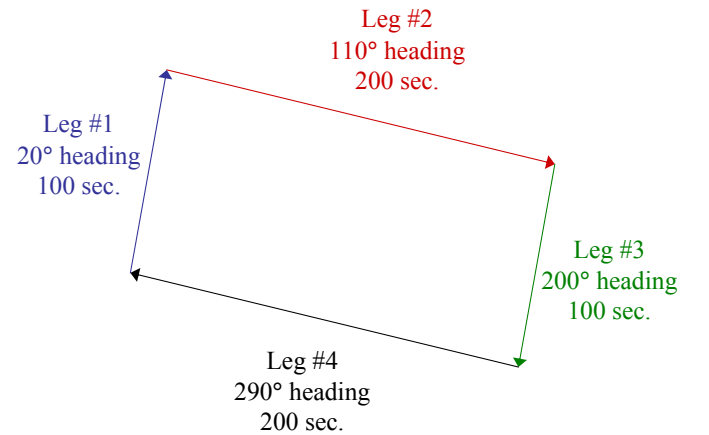


Fig. 4. Desired AUV mission corresponding to the results shown in Figs. 5 and 6.

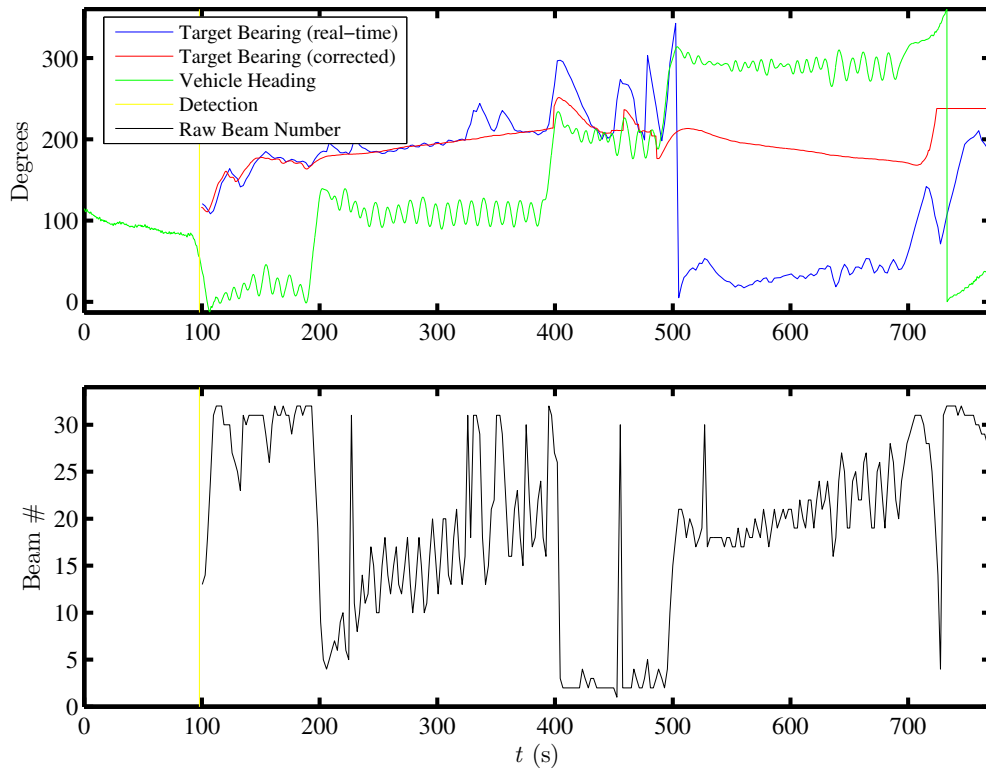


Fig. 5. Absolute target bearings computed both in real-time and then subsequently in post-processing after tuning parameters and modifying the beam tracking algorithm. This was done in order to increase performance and to compensate for an incorrect left/right ambiguity decision made in real-time for  $t > 500$  s.

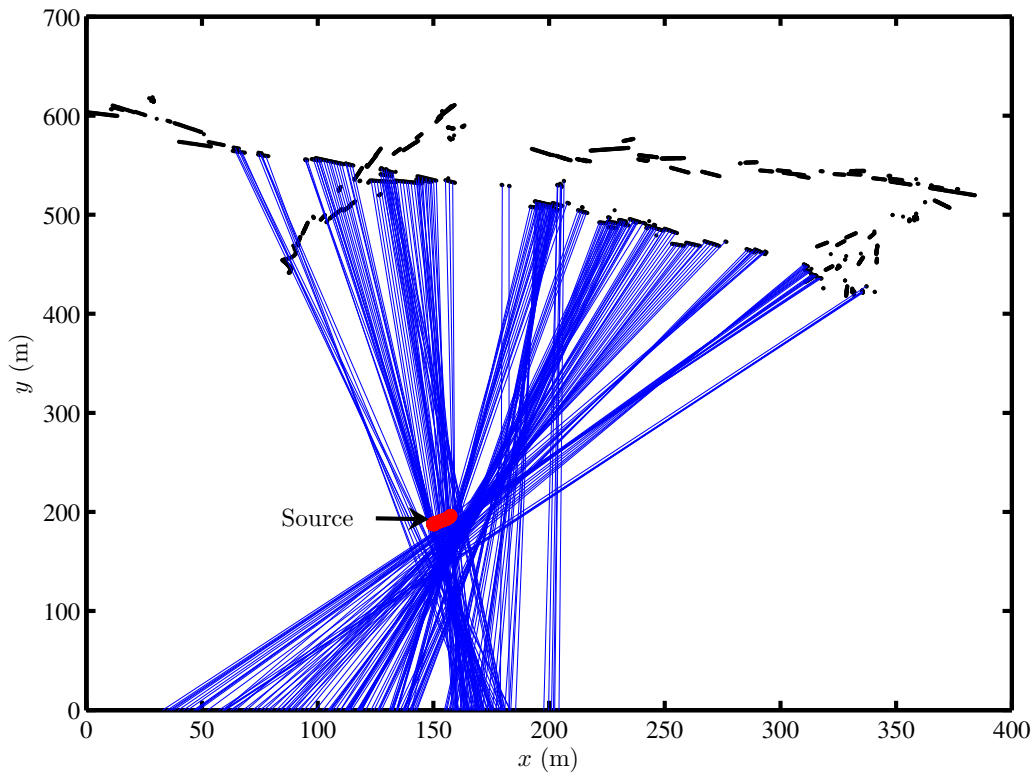


Fig. 6. Bearing lines originating from logged AUV navigation position data. This demonstrates consistency of bearing line triangulation with the logged GPS locations of the ship towing the source. Note that many of the outliers are due to jumps in the AUV navigation estimates.

#### IV. RESULTS

The bearing stabilization and tracking algorithm was successfully implemented and demonstrated during a sea trial in July 2005. A bandpass filtered pseudo-random gaussian noise source centered at 7 kHz with a 100 Hz bandwidth was towed from a stationary ship and used as a target. The source transducer was lowered via a cable from the aft of the ship. A 1024 point DFT was used prior to beamforming, with a 25% temporal window overlap. A Hanning function was used for both the temporal and frequency windowing operations. We set  $\alpha = 0.1$ , assumed a vertical angle of arrival of  $\phi = 0^\circ$  and used a bearing update time increment of  $\Delta t = 0.8$  sec. Eight acoustic channels were processed with inter-element spacing of 10 cm (this varied some due to the loss of a few faulty hydrophones). The tests were run in approximately 100 m of water.

Fig. 5 displays both the beam numbers and absolute target bearings computed in real-time for the mission with the desired rectangular AUV path shown in Fig. 4. This was one of our first trial runs. It is evident from the logged AUV location positions shown in Fig. 6 that the AUV “roughly” followed this desired path, but there are some significant errors both in the actual path taken and the measured AUV locations.

In post-processing, the beam tracking algorithm was modified and a coding error was resolved in order to compensate for an incorrect left/right ambiguity decision made for  $t > 500$  s. The resulting corrected absolute bearing measurements are included in Fig. 5.

Fig. 6 displays the reprocessed bearing lines originating from logged AUV navigation position data for  $510 \text{ s} < t < 707 \text{ s}$ , i.e. the fourth leg of the AUV mission. This data demonstrates the consistency of the bearing line triangulation with the logged GPS coordinates of the ship towing the source. In this case, the source was relatively stationary. Many of the bearing line outliers are due to jumps in the AUV navigation estimates. Improvements to this bearing stabilization and tracker will be implemented and tested at sea during August 2006.

#### V. CONCLUSION

Due to imperfections in vehicle control, an AUV can often undergo significant yaw and pitch oscillations while traversing through the water column. As a result, it is important to compensate for the vehicle motion when generating true bearing estimates. The beamforming, beam interpolation, bearing stabilization and tracking algorithms which were implemented to address these issues proved to be an effective method of measuring stabilized true target bearings in an underwater environment. They were successfully implemented on an intelligent AUV sonar sensor and tested during a subsequent sea trial.

Many levels of cooperation between the logical tracking sensor and the vehicle control system were developed and implemented. The beam tracking and stabilization algorithms are a component of a real-time software process which uses raw hydrophone data to produce stabilized true bearings. These

are then used by a separate real-time process to estimate a target track.

Improvements to these algorithms are currently being implemented for further sea trials in August 2006. Here they will form the core of an underwater passive acoustic target tracking system developed to autonomously and adaptively track moving contacts in both single-sensor and cooperative multi-sensor scenarios. The algorithm is also being extended to and tested with a flexible array towed by an AUV platform.

#### VI. ACKNOWLEDGEMENTS

The authors express gratitude to Arthur B. Baggeroer, Henrik Schmidt and David J. Battle for their valuable assistance. This work was supported by the PLUSNet program of the Office of Naval Research, Contract N00014-05-G-0106, Delivery Order 008.

#### REFERENCES

- [1] R. D. Ghen, D. C. Gilbert, and H. F. Jarvis, “Close Loop Contact Follower,” Analysis and Technology, Tech. Rep. P-2949-3-86, September 1986.
- [2] M. R. Benjamin, “Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control,” Ph.D. dissertation, Brown University, Providence, RI, May 2002.
- [3] M. R. Benjamin and J. Curcio, “COLREGS-Based Navigation in Unmanned Marine Vehicles,” in *AUV-2004*, Sebasco Harbor, Maine, June 2004.
- [4] M. Benjamin, J. Curcio, J. Leonard, and P. Newman, “Navigation of Unmanned Marine Vehicles in Accordance with the Rules of the Road,” in *International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.
- [5] M. Benjamin, M. Grund, and P. Newman, “Multi-objective Optimization of Sensor Quality with Efficient Marine Vehicle Task Execution,” in *International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.
- [6] D. Eickstedt and H. Schmidt, “A Low-Frequency Sonar for Sensor-Adaptive, Multi-Static, Detection and Classification of Underwater Targets with AUVs,” in *Oceans 2003*, San Diego, CA, 2003.
- [7] D. Eickstedt, M. Benjamin, H. Schmidt, and J. Leonard, “Adaptive Control of Heterogeneous Marine Sensor Platforms in an Autonomous Sensor Network,” in *press, Proceedings of the 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China, October 2006.
- [8] T. R. Benedict and G. W. Bordner, “Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations,” in *IRE Transactions on Automatic Control*, no. AC-6, July 1962, pp. 27–32.